

# PATENT ABSTRACTS OF JAPAN

(11)Publication number : **03-282934**

(43)Date of publication of application : **13.12.1991**

(51)Int.Cl.

**G06F 9/44**

(21)Application number : **02-081474**

(71)Applicant : **INTERNATL BUSINESS MACH CORP  
IBM**

(22)Date of filing : **30.03.1990**

(72)Inventor : **MURAJI NAGAKATSU  
HIRAGA RUMI**

## **(54) GRAPHICAL USER INTERFACE MANAGING DEVICE**

### **(57)Abstract:**

**PURPOSE:** To simply generate and control user interface by using a relational tables describing a relation between a display graphics part and a procedure module and a relation between respective graphic parts and module parts as objects.

**CONSTITUTION:** The base table 10 of a relational table describing the individual graphics part displayed on a display and the procedure module as an object, a relationship table 11 similarly describing the relation between the respective graphic parts and the relation between the respective module parts as the objects and a relational transition table 12 describing the internal transition of application as an object are stored in an object storing means. Then through a control program 13, a message is converted to the format of a relational table to refer to these tables 10 to 12 so that the user interface is simply generated and managed.



### **LEGAL STATUS**

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

## ④ 公開特許公報 (A) 平3-282934

⑤ Int.CI.<sup>5</sup>  
G 06 F 9/44識別記号  
330 Z内整理番号  
8724-5B

⑥ 公開 平成3年(1991)12月13日

審査請求 有 請求項の数 8 (全14頁)

⑦ 発明の名称 グラフィカル・ユーザ・インターフェース管理装置

⑧ 特願 平2-81474

⑨ 出願 平2(1990)3月30日

⑩ 発明者 連 永 勝 東京都千代田区一番町22-5 ホーマットオリエント701

⑪ 発明者 平賀 鑑 美 千葉県松戸市小金きよしが丘3-4-22 メゾン北小金  
502⑫ 出願人 インターナショナル・  
ビシネス・マシーン  
ズ・コーポレーションアメリカ合衆国10504、ニューヨーク州 アーモンク(番  
地なし)

⑬ 復代理人 弁理士 澤田 俊夫

## 明細書

## 1. 発明の名称

グラフィカル・ユーザ・インターフェース  
管理装置

## 2. 特許請求の範囲

(1) 表示装置上に表示される個々のグラフィックス部品と、1または1群の手続モジュールと、上記グラフィックス部品および手続モジュールの間の関係、上記グラフィックス部品どうしの間の関係および上記手続モジュールどうしの間の関係とをそれぞれオブジェクトとして記述する関係型のテーブルをストアするオブジェクト記憶手段と、

上記オブジェクトに対するメッセージを上記関係型のテーブルのフォーマットに変換し、上記関係型のテーブルに対して所定の検索を実行し、上記検索の結果に応じた手續モジュールの呼び出しを行う制御手段とを有することを特徴とするグラフィカル・ユーザ・インターフェース管理装置。

(2) 上記グラフィックス部品にはパネルのレイアウトを構成する縦や矢印等のグラフィックス部品と、パネル自体とが含まれる請求項(1)記載のグラフィカル・ユーザ・インターフェース管理装置。

(3) 上記手続モジュールはアプリケーション本体への加えられるイベント・ストリームを含む請求項(1)記載のグラフィカル・ユーザ・インターフェース管理装置。

(4) 上記手続モジュールはアプリケーション本体へ加えられる機能呼び出しを含む請求項(1)記載のグラフィカル・ユーザ・インターフェース管理装置。

(5) 上記手続モジュールは検索ルーチンを含む請求項(1)記載のグラフィカル・ユーザ・インターフェース管理装置。

(6) 表示装置上に表示される個々のグラフィックス部品と、1または1群の手續モジュールと、上記グラフィックス部品および手續モジュールの間の関係、上記グラフィックス部品どうしの間の

関係および上記手続モジュールどうしの間の関係とをオブジェクトとして記述する関係型のテーブルをストアするオブジェクト記憶手段と、

アプリケーションの内部状態の遷移をオブジェクトとして記述する関係型のテーブルをストアする遷移オブジェクト記憶手段と、

上記オブジェクトに対するメッセージを上記関係型のテーブルのフォーマットに変換し、上記関係型のテーブルに所定の検索を実行し、上記検索の結果に応じた手続モジュールの呼び出しを行う制御手段とを有することを特徴とするグラフィカル・ユーザ・インターフェース管理装置。

(7) 表示装置上に表示される個々のグラフィック部品と、1または1群の手続モジュールとを基本オブジェクトとして記述する関係型のテーブルをストアする基本オブジェクト記憶手段と、

上記基本オブジェクトどうしを関係付ける複合オブジェクトを記述する関係型のテーブルをストアする複合オブジェクト記憶手段と、

アプリケーションの内部状態の遷移をオブジ

クトとして記述する関係型のテーブルをストアする遷移オブジェクト記憶手段と、

オブジェクトに対するメッセージを上記関係型のテーブルのフォーマットに変換し、上記関係型のテーブルに所定の検索を実行し、上記検索の結果に応じた手続モジュールの呼び出しを行う制御手段とを有することを特徴とするグラフィカル・ユーザ・インターフェース管理装置。

(8) アプリケーションに依存しないグラフィカル・ユーザ・インターフェースのオブジェクトを記述する関係型のテーブルをストアする第1のオブジェクト記憶手段と、

アプリケーションに依存するグラフィカル・ユーザ・インターフェースのオブジェクトを記述する関係型のテーブルをストアする第2のオブジェクト記憶手段と、

アプリケーションの内部状態の遷移をオブジェクトとして記述する関係型のテーブルをストアする遷移オブジェクト記憶手段と、

オブジェクトに対するメッセージを上記関係型

のテーブルのフォーマットに変換し、上記関係型のテーブルに所定の検索を実行し、上記検索の結果に応じた手続モジュールの呼び出しを行う制御手段とを有することを特徴とするグラフィカル・ユーザ・インターフェース管理装置。

### 3. 発明の詳細な説明

#### A. 産業上の利用分野

この発明はグラフィカル・ユーザ・インターフェース管理装置に関し、とくに関係型のテーブルを用いて簡単にユーザ・インターフェースのオブジェクトを生成し、管理できるようにしたものである。

#### B. 従来の技術

オブジェクト指向によるプログラム構築は現在広まりつつある。

オブジェクト指向言語によるプログラム構築の特徴は、第一にデータ抽象の手法を用いる点にある。ここでデータ抽象とは、データ表現とデータに対するオペレーションとをひとまとめにし、データへのアクセスを与えられたオペレーション

のインターフェースのみから許すプログラムの書き方である。プログラム中のいくつかのデータは、共通のデータ表現とインターフェースをもって表される。オブジェクト指向言語では、共通するデータを表す枠組みを「クラス」と呼ぶことが多い。オブジェクトは、クラスで代表される構造を持つデータの実体である。アプリケーションは、複数のクラスの定義とオブジェクトの生成、オブジェクトへのオペレーションとにより書かれる。オブジェクト指向言語によるプログラム構築の第二の特徴は、クラスが階層を持ち、上下クラス間では、オペレーションやデータ表現の継承が行われるという点である。

これらの特徴により、より高レベルのデータ・モデル化ヒデータ型の共有、再利用を可能にしている。

しかし、クラスの定義とその利用には、オブジェクト指向言語を用いなければならず、データ型の共有・再利用も、オブジェクト指向言語において同じくしている。したがって、既存のファイルは

もちろん、異なるオブジェクト指向言語で定義されたクラスの利用も不可能である。

近年オブジェクト指向言語により、既々のユーザー・インターフェース管理装置が試作されている。P.A.Szekely およびB.A.Myersによるユーザ・インターフェース・ツール・キット "Coral" (A User Interface Toolkit Based on Graphical Objects and Constraints, DOPSLA '88 Conference Proceedings pp.36-45) は、ウインドウ中のグラフィカル・オブジェクトを作り出す装置である。

Coralは、LISPを基本としたオブジェクト指向言語(CLOS)により書かれている。ユーザにとって Coralの特徴は、

1. グラフィカル・オブジェクト定義のための宣言言語を定め、ユーザによるオブジェクト定義ができるようにしたこと、
2. オブジェクト間の制約設定のための手続き言語を定め、ユーザによる制約設定を可能にしたことである。

ト指向言語に「関係」の概念がないからである。したがって設計時に特別な変数やクラスの導入を考えなければならない。

また特徴3のようなオブジェクト・アクセスの機能は、オブジェクトをデータ保管場所と考えれば、オブジェクトの外部にあるのが、データベース・システムの考え方にも沿い、自然である。

なおこの発明と関連する他の先行技術としては特開平1-229321号公報がある。この先行技術は表示しているオブジェクトの位置情報をテーブルとして保持するユーザ・インターフェース作成ツールである。しかしこの先行技術はテーブルを関係型として複数のテーブルを検索することは示唆していない。またグラフィカル・オブジェクトや手続オブジェクトを直接テーブルの形で生成、管理するものではない。

#### C. 発明が解決しようとする課題

この発明は以上の事情を考慮してなされたものであり、オブジェクト指向言語を用いることな

る。ユーザが新たなグラフィカル・オブジェクトを作成することは、Coralが基礎としているCLOSとCoralの定めた宣言言語の理解が、また、オブジェクト間の制約の記述は、CLOSと手続き言語の習得が前提となる。

ユーザ・インターフェース管理装置の構築という点からのCoralの特徴は、

1. グラフィカル・オブジェクトの表示部分とそれが起動する実行部分を分離するために特別の変数を用意したこと、
2. 集合体クラスを用意し、複数のグラフィカル・オブジェクトをまとめて扱いやすくしたこと、
3. 表示されているグラフィカル・オブジェクトを指す入力に対して、オブジェクトを識別するプロシジャーとオブジェクト中のデータの値の変更などは、オブジェクト内に定義されていることである。

上述の特徴1および2のように特別な変数やクラスを用意しなければならないのは、オブジェク

ト指向言語に「関係」の概念がないからである。したがって設計時に特別な変数やクラスの導入を考えなければならない。

#### D. 課題を解決する手段

この発明のグラフィカル・ユーザ・インターフェースのオブジェクト管理装置では以上の課題を解決するために、表示装置上に表示される個々のグラフィックス部品と、1または1群の手続モジュールとを基本オブジェクトとして記述する関係型のテーブルをストアする基本オブジェクト記憶手段と、上記基本オブジェクトどうしを関係付ける複合オブジェクトを記述する関係型のテーブルをストアする複合オブジェクト記憶手段と、アプリケーションの内部状態の遷移をオブジェクトとして記述する関係型のテーブルをストアする遷移オブジェクト記憶手段と、オブジェクトに対するメッセージを上記関係型のテーブルのフォーマットに変換し、上記関係型のテーブルに所定の検索を実行し、上記検索の結果に応じた実行モジュールの呼び出しを行う制御手段とを有するよ

うにしている。

#### E. 実施例

以下、この発明をプログラミング支援装置のユーザ・インターフェースのオブジェクトの管理に適用した実施例について説明する。このプログラミング支援装置は1または複数のアプリケーションをカスタマイズして新たな1つのアプリケーションを作成するのを支援するものである。

まずこの実施例のユーザ・インターフェースのオブジェクトの管理のアーキテクチャについて概説しよう。

第1図はこのオブジェクト管理アーキテクチャを示し、この第1図においてオブジェクト管理アーキテクチャはベース・テーブル群10、リレーションシップ・テーブル群11、遷移テーブル群12およびコントロール・プログラム13からなっている。ベース・テーブル群10は1または複数のベース・テーブルを含み、ベース・テーブルは基本的なオブジェクトを記述する。基本的なオブジェクトは、汎用的にアプリケーションの

セマンティクスに依存しないものである。例として、長方形や矢印などの基本的なグラフィカル・オブジェクトのための、大きさや色の属性をもつテーブルをあげることができる（第14図のボックス・テーブルおよび矢印テーブル）。またグラフィカル・オブジェクトに関連して実行される手続きを定義するテーブルを挙げができる（第13図の手続き定義テーブル）。オブジェクト指向言語におけるクラス変数とメソッド（オペレーション）は、このオブジェクト管理アーキテクチャでは、どちらもテーブルの属性中に定義される。オペレーションの種類は、計算やオブジェクトの表示をするためのエクゼキューション・モジュールまたは、オブジェクトの検索をするためのオペレーション・モジュールの二種類がある。

リレーションシップ・テーブル群11は1または複数のリレーションシップ・テーブルを含み、これらリレーションシップ・テーブルは、ベース・テーブルで定義されたオブジェクトをもとにアプリケーション・セマンティクスを満足する複

雑なオブジェクトを定義するために用いられる。複数のグラフィカル・オブジェクトで構成される画面は、リレーションシップ・テーブル中で定義される（第15図のパネル定義テーブル）。表示されたグラフィカル・オブジェクトにナビゲーションの機能を加えるときもこのテーブルを用いる（後述の表1、2および3の検索式参照）。アプリケーションに依存しないオブジェクトとアプリケーションに依存するオブジェクトをそれぞれベース・テーブルおよびリレーションシップ・テーブルとして分離しているのでユーザ・インターフェースの変更や複数のユーザ・インターフェースを設計するのに便利である。

遷移テーブル群12は1または複数の遷移テーブルを含んでいる。遷移テーブルはアプリケーションの内部状態の遷移やパネルの遷移を表すものである（第21図のパネル遷移テーブル参照）。ユーザのアクションに対して生ずる次状態への遷移や次のパネルの表示は、このテーブルを参照して行なわれる。

コントロール・プログラム13は上述の関係型のテーブルとその外部との間のインターフェースを提供するものである。たとえばシステム・キー中のウィンドウ・メッセージをテーブルのフォーマットに変換する。そのために変換テーブル（第18図のウィンドウ・メッセージ・テーブル）を保持している。またコントロール・プログラム13は関係型のテーブルに対する検索機能も有している（後述の表1、2および3の検索式参照）。

つぎにプログラミング支援装置の具体例に沿ってこの発明を詳述しよう。なおこの発明は具体的にはプログラミング支援装置のユーザ・インターフェース・イネーブラ UIE 103 のユーザ・インターフェースのオブジェクト管理に適用されている。

第1図はこのプログラミング支援装置100が実現されるシステムの全体構成を示している。この第1図において、プログラミング支援装置100はアプリケーション・ロジック・ネーブラ（ALE）

101、カスタマ・プログラミング・ファシリティ (CPF) 102およびユーザ・インターフェース・イキーブラ 103からなっている。この実施例のプログラミング支援装置 100 は IBM PS/55 (米国インターナショナル・ビジネス・マシーン (IBM) 社の商標) ワークステーション 104 上で構成し、オペレーティング・システムならびにウィンドウの管理をおこなうプレゼンテーション・マネージャからなるシステム・プログラム (以下プレゼンテーション・マネージャ/オペレーティング・システムと呼ぶ) 105 上にソフトウェアとして実現されている。

ALE 101は既存のアプリケーションの機能を提供するものであり、実際にはプレゼンテーション・マネージャ/オペレーティング・システム 105 上の 1つまたは 1組のアプリケーションである。ALE 101 はより多くの機能を提供することが望まれ、通常は汎用のオフィス・アプリケーションたとえばスプレッド・シート、ワードプロセッサ、データベース管理、計算機能、ビジネス・チャートのパッケージである。もちろん必要に応じて特別な用途のアプリケーション、たとえば知識ベース・システム用のカーネルや意志決定支援システムの推論エンジンを採用してもよい。具体的な例ではALE 101として IBM社のプレゼンテーション・マネージャ上の複数のアプリケーションを用いる。これらアプリケーションとしては商業プログラマが作成した汎用のオフィス・アプリケーションたとえばスpreadsheet・シート、ワードプロセッサ、データベース管理、計算機能、ビジネス・チャートのパッケージが用意されている。さらにこれら複数のアプリケーションの間ではデータを交換できるようになっている。したがって多種多様な機能の提供が約束される。なおプレゼンテーション・マネージャのかわりに IBM 社の SMART シリーズ ( IBM 社の商標) のウィンドウ・システムを用いる場合には、SMART シリーズ用の複数のアプリケーションを利用できる。SMART シリーズは複数のアプリケーションの実行を支援するウィンドウ・ベースのシステムである。

UIE 103はユーザ・インターフェースのレイアウト (画面) を編集するグラフィックス・エディタである。この UIE 103 によって新たに生成されたユーザ・インターフェースのレイアウトのグラフィカル・オブジェクトおよび後述するように CPF 102 により生成された手続モジュール (ロジック・モジュール) のオブジェクト (後述するイベント・ファイルおよびストーリ) とを管理するのに、この発明が適用されている。

CPF 102 はこの実施例のプログラミング支援装置の要部をなすもので、この支援のもとユーザは ALE 101のアプリケーションをカスタマイズして新しいアプリケーションを構成でき、しかも新しいアプリケーションと、新たなユーザ・インターフェースのレイアウトとを関係付けるようになっている。

ALE 101 に対して、CPF 102 はあたかも存在していないかのように見えるが、実際はプレゼンテーション・マネージャ/オペレーティング・システム 105 とALE 101のアプリケーションとの情報

トのパッケージである。もちろん必要に応じて特別な用途のアプリケーション、たとえば知識ベース・システム用のカーネルや意志決定支援システムの推論エンジンを採用してもよい。具体的な例ではALE 101として IBM社のプレゼンテーション・マネージャ上の複数のアプリケーションを用いる。これらアプリケーションとしては商業プログラマが作成した汎用のオフィス・アプリケーションたとえばスpreadsheet・シート、ワードプロセッサ、データベース管理、計算機能、ビジネス・チャートのパッケージが用意されている。さらにこれら複数のアプリケーションの間ではデータを交換できるようになっている。したがって多種多様な機能の提供が約束される。なおプレゼンテーション・マネージャのかわりに IBM 社の SMART シリーズ ( IBM 社の商標) のウィンドウ・システムを用いる場合には、SMART シリーズ用の複数のアプリケーションを利用できる。SMART シリーズは複数のアプリケーションの実行を支援するウィンドウ・ベースのシステムである。

のやり取りの制御をおこなう。

この実施例のプログラミング支援装置 100 はカスタマイズを行うためのプログラム・モード (第3図) とカスタマイズされた環境を実現化させるための実行モード (第4図) をもつ。プログラム・モードで新たなユーザ・インターフェースのオブジェクトが関係型のテーブルの形態で生成され、実行モードでそれらオブジェクトの関係型のテーブルを操作してカスタマイズしたプログラムを実行する。なおプログラム・モードと実行モードとはたとえば、メニュー やキーによって切り換えるように実現することができる。

第3図はプログラム・モードのフローを示す。プログラム・モードは、実行モードでの実行に先立って、アプリケーション上で実行すべき事柄をあらかじめ定義するためのモードである。第3図に示すように、このモードでは、ユーザが実際にアプリケーションに対する操作を行い、この操作が記録される (S11)。このような仕組をレコードと呼び、一連の操作はイベント・ファイル

## 特開平3-282934 (6)

とよばれるデータとして保存される。保存された1または複数のイベント・ファイルはループや分岐などの制御構造の記述が可能なストーリ・エディタ 107(第5図)によってストーリという形式のデータを作成することも可能である(S12)。イベント・ファイル、ストーリはともに単独で実行することが可能なデータである。つぎに、実行する内容を指示するためのユーザ・インターフェイスの定義をおこなう(S13)。ユーザ・インターフェースはグラフィカル・オブジェクトを組み合わせて作られるが、最後にリンクという操作でイベント・ファイルまたはストーリに対応させる(S14)。

なお各ステップの詳細については第5図以後の画面を参照して後に説明する。

第4図は実行モードのフローを示す。実行モードは、プログラム・モードでカスタマイズされたユーザ・インターフェイスから記録された操作を再実行するモードである。第4図に示すように、このモードにおいては、ユーザが画面上のグラ

フィカル・オブジェクトをクリックすると(S21)、このグラフィカル・オブジェクトに対応するイベント・ファイルやストーリが起動される(S22)。

つぎにこの実施例のプログラミング支援装置の各部を順をつけて説明する。

第5図はCPF 102の構成を示している。この第5図において CPF 102 はイベント・レコーダ 106、ストーリ・エディタ 107、リンク 108からなっている。イベント・レコーダ 106はプログラム・モード時にユーザが実際に入力する一連のイベントをイベント・ファイルとして記録し、実行モード時にそれを再生するものである。ストーリ・エディタ 107は、記録されたイベント・ファイルをユーザが組合せて1組のイベント・ファイルすなわちストーリを生成するのを支援する。リンク 108はイベント・ファイルまたはストーリを、新たに作成したユーザ・インターフェースのグラフィカル・オブジェクトに関係付けるものである。

第6図はイベント・レコーダ 106の詳細を示す。この第5図において、イベント・レコーダ 106はイベント記録部 109、イベント・ファイル記憶部 110 およびイベント再生部 111 からなっている。

イベント記録部 109 はアプリケーション・プログラム 112 がユーザの振舞いを知るために用いるアプリケーション・プログラム・インターフェースを監視し、制御する機能を持っている。この側ではプレゼンテーション・マネージャ 105a が提供するイベント・キュー 105b からの情報も監視、制御の対象とする。すなわちアプリケーション・プログラム 112 がユーザの振舞いを知るためにイベント・キュー 105b からイベントを取りごとに、イベントを横取りし、一連のイベントをイベント・ファイルとしてイベント・ファイル記憶部 110 に記憶させる。具体的な例では、プレゼンテーション・マネージャ 105a により提供されるインプット・フックという機能によってイベントの横取りを行っている。このインプッ

ト・フックの詳細については IBM Operating System/2 Programmer's Toolkit Version 1.1 programming Guide (Operating System/2は IBM 社の商標) pp.11-18~11-19 を参照されたい。イベント・ファイル記憶部 110 に記憶されたイベント・ファイルは第13図に示す手続定義テーブル(ベース・テーブル)と呼ぶ関係型のテーブルの形態で管理される。なお第6図で 105c はオペーティング・システムである。

第7図は記録されるイベントの構造(A)並びに一例(B)を示す。ここでは、マウス、キーボードによる直接的なイベント並びにメニューやウィンドウ及びアプリケーションの起動などのシステムによるイベントが扱われる。イベントの情報はウィンドウのID(Window Handle) 113a、イベントの種類(Message) 113b、タイム・スタンプ(Timestamp) 113c、メッセージごとに意味が異なるそのほかのパラメータ(Parameters...) 113d からなっている。

第6図においてイベント再生部 111 は再生

特開平3-282934(7)

モード時にイベント記憶部110に記憶されているイベントを再生するものである。イベント再生部111はアプリケーション・プログラム112がイベントをイベント・キューレコード105bから取ろうとしたとき、イベント・ファイル記憶部110にあるイベントをあたかもイベント・キューレコード105bから来たイベントのようにしてアプリケーション・プログラム112にわたす。具体的な例ではアプリケーション間の通信機能などを実現するために用意されている機能の一つ、すなわちメッセージを送るための機能(*WinSendMsg*)を用いてこの再生を行う。この*WinSendMsg*の機能の詳細についてはIBM Operating System/2 Programmer's Toolkit Version 1.1 programming Guide (Operating System/2はIBM社の商標) sp.8-6~3-8を参照されたい。

第8図はストーリ・エディタ107のユーザ・インターフェース画面の一例を示す。第8図に示すようにストーリ・エディタ107はイベント・リスト107a、ストーリ・ボード107bおよびコマン

ド107cの各領域を提供し、視覚的にイベントを編集できるようにしている。ストーリ・エディタ107は、1または複数のイベント・ファイルをひとまとめにして扱うことを可能にする。登録されたイベント・ファイルはイベント・リスト107aによって参照できる。これらのイベント・リスト107aはコマンド107cにしたがってストーリ・ボード107bの上に生成される。ストーリ・エディタ107のもっとも重要な機能は、1または複数のイベント・ファイルを組み合わせて実行可能なさらに大きな記録を作れることである。また、ストーリ中ではコマンド領域のコマンド107cを用いて制御構造の説明を行うことができる。以下ではストーリをイベント・ファイルと一括して手続モジュールと呼ぶことにする。ストーリもイベント・ファイルと同様にイベント・ファイル記憶部110に第13図に示す関係型のテーブルの形態で保持されている。

第5図のリンクはユーザ・インターフェースへのオペレータ（たとえばマウス・クリック）に応

答して手続モジュールを起動するようにユーザ・インターフェースのグラフィック・オブジェクトと手続モジュールとを連結するものである。グラフィカル・オブジェクトと手続モジュールとの関連付けは後述するようにUIE103に用いてユーザが行ない、対応結果は第16図のオブジェクト・オペレーション・テーブルおよび第17図の手続起動テーブルの各関係型のテーブル（リレーションシップ・テーブル）ならびに後述の表1、表2および表3の検索式として表わされ保持される。これら関係型のテーブルや検索式の作成、保持はUIE103を用いて行なわれる所以、それらの詳細はUIE103の説明に譲る。

つぎにUIE103について説明する。UIE103はユーザ・インターフェイスを定義するための一種のグラフィックス・エディタである。表示部分は一般的なWYSIWYG(What you see is what you get)方式のグラフィックス・エディタと変わることろはなく、ユーザは自由な图形をユーザ・インターフェースの表示に用いることができる。この

グラフィックス・エディタの機能部分がレイアウト・エディタ114であり、このレイアウト・エディタ114を用いて、ユーザはこれから利用する新たなユーザ・インターフェースの見かけを自由に定義することができる。

第10図はレイアウト・エディタ114で作成したレイアウトの一例を示す。この例において118-120は文字列と矩形をグループ化して構成したグラフィカル・オブジェクトであり、それぞれのグラフィカル・オブジェクトはレイアウト・オブジェクトと呼ばれ、手続モジュールと対応付けられている。この第10図は、レポートを作成して印刷するまでのオフィスにおける作業の過程をUIE103で作成した例を示すが、矢印121は、ここでは手続モジュールに対応づけられたものではなく単なる作業手順の目安となっている（必要があれば何らかの手続モジュールを対応させることも可能である）。

これらの图形情報の定義はレイアウト・ファイル115（第9図）という形式で扱われ、ユー

## 特開平3-282934 (8)

ザ・インターフェイスの保存に用いられる。レイアウト・ファイル 116 は具体的には第14図のボックス・テーブルおよび矢印テーブル(ベース・テーブル)ならびに第15図のパネル定義テーブル(リレーションシップ・テーブル)を含んでいる。.

つぎに、これらレイアウト・オブジェクトに、手続モジュールを対応付ける方法について述べる。第11図はリンク作業を行っている例であり、第12図はその手順を示す。第11図および第12図において、最初にリンクしたいグラフィカル・オブジェクト 122 をクリックする(S31)。これによって第16図に示すようにグラフィカル・オブジェクトと遷移とが関係付けられる。すなわちグラフィカル・オブジェクト 122 をクリックするたびに第16図のテーブルに新たなタブルが加えられる。なお遷移はアプリケーションの内部状態およびパネルの遷移である。またこのときのクリックによって対象がハイライトされ、ユーザが確認できるようになっている(S32)。なお第16

図はどのグラフィカル・オブジェクトに何のオペレータが加えられたときにどのような遷移が起こるのかを示しており、この例ではオペレータはマウス・クリックがデフォルト・パラメータになっている。

つぎに、UIE 103 のメニュー 123 からリンク・アクションを選ぶと(S33)、ストーリカイベントかを選択するメニュー 124 が出てくる(S34)。この例ではストーリを選択する(S35)。次に、ストーリのリストのダイアログボックス 125 があらわれる(S36)。ここリストから必要な手続モジュールをクリックによって選択すると(S37)、第17図に示すように遷移と起動手續モジュールとが対応付けられる。こうしてリンクが確定する(S38)。

また、レイアウト・オブジェクトからのリンクを他の手續モジュールに張り替える際も同様の操作でリンクがおこることが可能である。

なおレイアウト・オブジェクトどうしを関係付けて、1つのレイアウト・オブジェクトに関連す

る手續モジュールの実行に統けて他のレイアウト・オブジェクトに関連する手續モジュールを実行させれるようにもできる。

このようにして作成したユーザ・インターフェースはたとえば第10図のような見かけを有し、ユーザの操作の応じて手續モジュールを呼び出し、アプリケーションに順次イベントを供給し、ユーザが希望したタスクを自動実行する。すなわちウインドウ・システムとのインターフェース、およびテーブル中のオブジェクト検索の開始のためにコントロール・プログラムが動作する。例えば、マウス・クリックがオブジェクトObj002 上で起こったとき、コントロール・プログラムのオペレーション・シーケンスは、以下のとおりである。

1. ウィンドウ・メッセージを第18図のウィンドウ・メッセージ・テーブルのエンティティに変換する。
2. マウス・クリックの対象となったグラフィカル・オブジェクトをさがし、オブジェクトIDとし

てObj002をもつエンティティが選ばれる。

このときのコントロール・プログラムの機能はつぎのような擬似コードで表わされる。

(式1)

```
SELECT オブジェクトID FROM  
ボックス・テーブルまたは矢印テーブル  
WHERE ウィンドウ・メッセージ・テーブル  
のパラメータ1がパネル定義テーブル。  
位置とボックス・テーブル、サイズで示さ  
れる矩形の中にある
```

3. 選ばれたグラフィカル・オブジェクトと、入力の組み合わせは、アプリケーションの遷移を起こす可能性があるので、オブジェクト・オペレーション・テーブルにObj002が存在するかどうかを調べる。そしてObj002に対するマウス・クリックは、遷移Tr002を引き起こすことがわかる。

このときのコントロール・プログラムの機能はつぎのような擬似コードで表わされる。

(式2)

```

SELECT 移移ID FROM オブジェクト
 ト・オペレーション・テーブル
WHERE オブジェクト・オペレーション・
  テーブル、オペレータ=ウィンドウ・メッセージ・
  テーブル、メッセージID
AND オブジェクト・オペレーション・
  テーブル、オブジェクトID=式1で選ばれたオブジェクトID

```

4. 選ばれた遷移が何を実行するかを手続起動テーブルよりさがす。そしてTr002は、手続Pro002を起動する。

このときのコントロール・プログラムの機能はつぎのような疑似コードで表わされる。

(式3)

```

SELECT 起動手続 FROM 手続起動
  テーブル
WHERE 手続起動テーブル、遷移ID=式2で選ばれた遷移ID

```

CALL 選ばれた手続の起動

ルで実行のボタンをクリックすると印刷の手続が呼び出されるようになっている。なおこの第19図の例では第20図のパネル・テーブルおよび手続定義テーブルが用意され、また第21図のパネル遷移テーブルおよび手続起動テーブルが用意される。

#### F. 発明の効果

以上説明したように、この発明によれば関係型のテーブルを採用することによりユーザ・インターフェースのオブジェクトを簡易に生成、管理するようことができる。

#### 4. 図面の簡単な説明

第1図はこの発明のユーザ・インターフェース管理装置の一実施例を概念的に示すブロック図、第2図は第1回のユーザ・インターフェース管理装置を採用しているプログラミング支援装置を全体として示すブロック図、第3図は第2回のプログラミング支援装置のプログラム・モードを説明するフローチャート、第4図は第2回のプログラミング支援装置の実行モードを説明するフローチ

なお以上の例では第16図および第17図から明らかのように、クリック対象のグラフィカル・オブジェクトと起動対象の手続モジュールとが一对一に対応するので、手続モジュールの呼び出しに関する記述はユーザ・インターフェースの内部状態の遷移を個別に記述、管理する必要性は少ないが、ユーザ・インターフェースのオブジェクトの生成、管理の点からは遷移を定義してユーザ・インターフェースの動的なシナリオを設計することは重要である。また第19図に示すように、遷移と起動対象手続モジュールとは一对一に対応するけれど、クリック対象のグラフィカル・オブジェクトと起動対象の手続モジュールとが必ずしも一对一に対応しない場合には、ユーザ・インターフェースの内部状態の遷移を個別に記述、管理する必要がある。なお第19図では手続パネルのウィンドウで印刷のボックスがクリックされると印刷パネルのウィンドウが表示され印刷パラメータを入力できるようになっており、この印刷パネ

ルで実行のボタンをクリックすると印刷の手続が呼び出されるようになっている。なおこの第19図の例では第20図のパネル・テーブルおよび手続定義テーブルが用意され、また第21図のパネル遷移テーブルおよび手続起動テーブルが用意される。

F. 発明の効果

以上説明したように、この発明によれば関係型のテーブルを採用することによりユーザ・インターフェースのオブジェクトを簡易に生成、管理するようができる。

4. 図面の簡単な説明

第1図はこの発明のユーザ・インターフェース管理装置の一実施例を概念的に示すブロック図、第2図は第1回のユーザ・インターフェース管理装置を採用しているプログラミング支援装置を全体として示すブロック図、第3図は第2回のプログラミング支援装置のプログラム・モードを説明するフローチャート、第4図は第2回のプログラミング支援装置の実行モードを説明するフローチ

マート、第5図は第2回のカスタマ・プログラミング・ファシリティ (CPF) の構成を示すブロック図、第6図は第5図のイベント・レコーダの詳細を示すブロック図、第7図はイベント・レコーダで扱うイベントを説明する図、第8図は第5図のストーリ・エディタの動作を説明する図、第9図は第2回のユーザ・インターフェース・イネーブラ (UIE) の構成を示す図、第10図、第11図および第12図は第2回のユーザ・インターフェース・イネーブラの動作を説明する図、第13図から第18図はこの実施例の関係型のテーブルを示す図、第19図から第21図は上述した実施例の変形例を説明する図である。

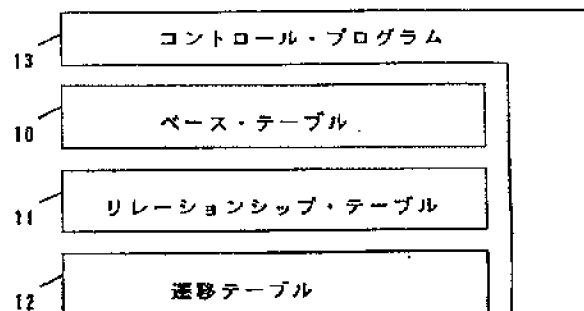
10—ベース・テーブル群、11—リレーションシップ・テーブル群、12—遷移テーブル群、13—コントロール・プログラム、100—プログラミング支援装置、101—アプリケーション・ロジック・イネーブラ (ALE)、102—カスタマ・プログラミング・ファシリティ (CPF)、103—ユーザ・

特開平3-282934 (10)

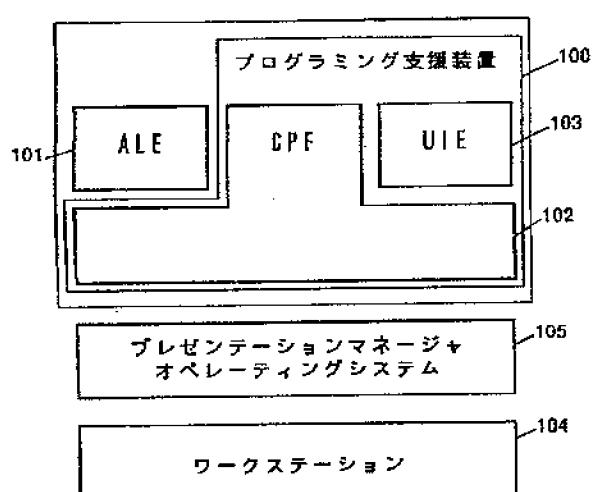
インターフェース・イネーブラ (UIE)。

出願人 インターナショナル・ビジネス・  
マシーン・コーポレーション

復代理人 弁理士 神田 優夫

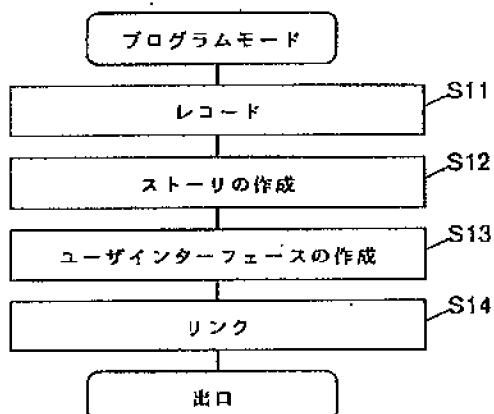


第1図 オブジェクト管理アーキテクチャ



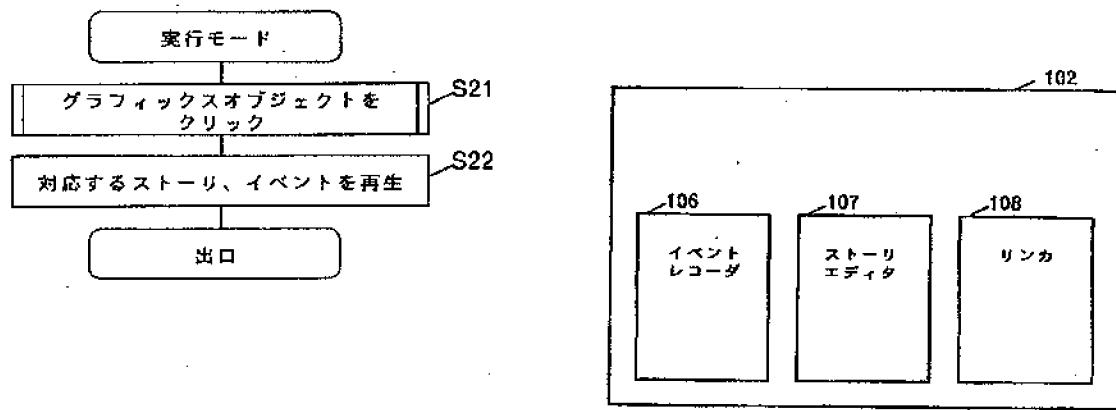
システムの全体構成

第2図



プログラムモードのフロー

第3図

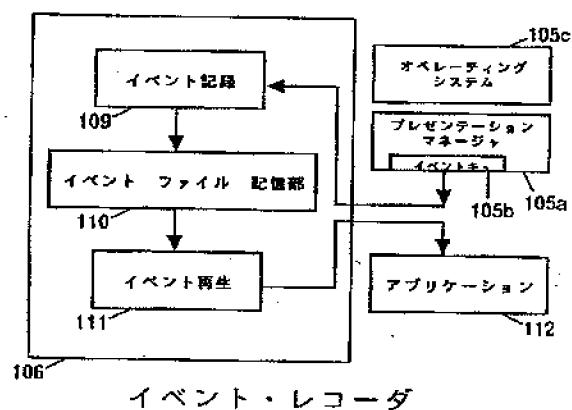


CPFの構成

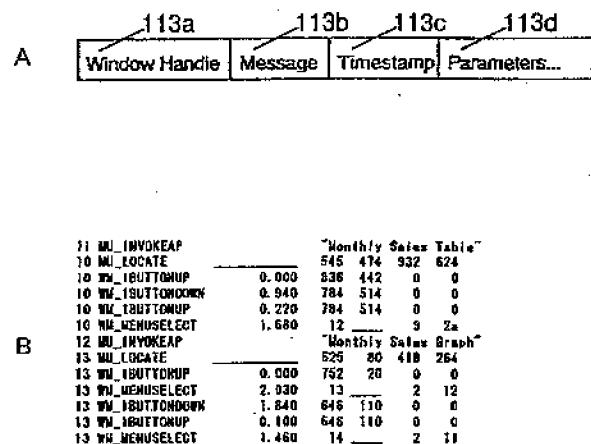
実行モードのフロー

第4図

第5図

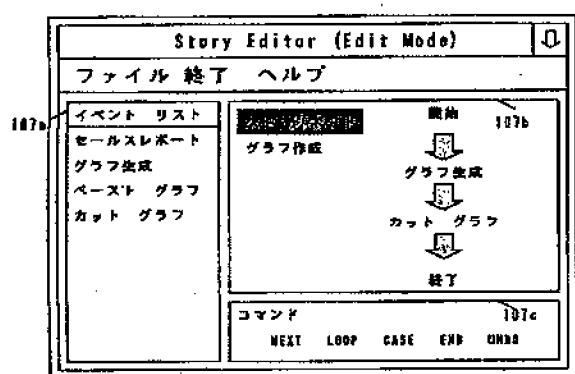


第6図



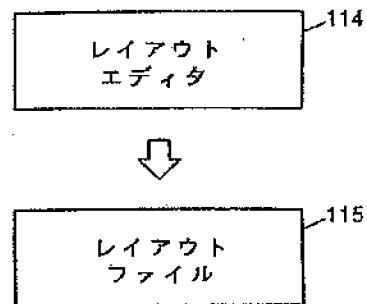
イベントの例

第7図



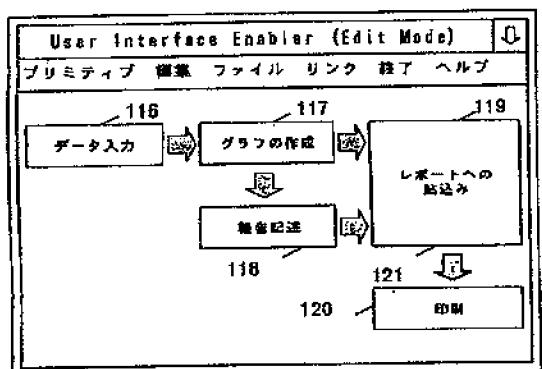
ストーリ・エディタ

第8図



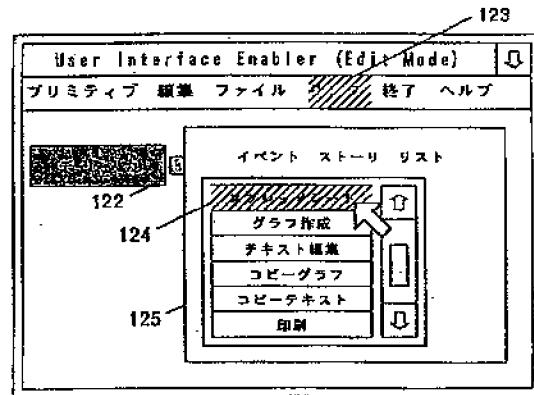
ユーザ・インターフェース・イネーブラの概要

第9図



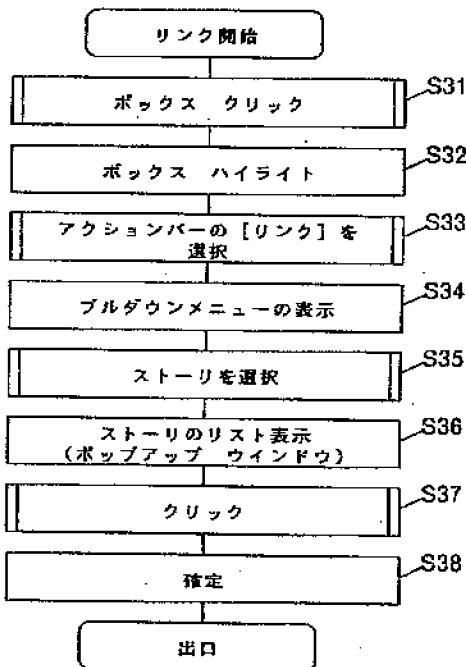
レイアウトの例

第10図



リンクの操作の例

第11図



## リンク操作の例

第12図

手続キ定義テーブル

手続キID	名前
Pro001	StopData
Pro002	MakeRpt
Pro003	WriteRpt
Pro004	PasteRpt
Pro005	PrintRpt

第13図 イベント・ファイルのテーブル  
(ベース・テーブル)

ボックス・テーブル

オブジェクトID	色	ラベル	サイズ
Obj001	緑	データ入力	200,50
Obj002	緑	グラフの作成	200,50
Obj003	緑	報告記述	200,50
Obj004	緑	レポートへの貼込み	200,200
Obj005	黄	印刷	200,50

矢印テーブル

オブジェクトID	方向	サイズ
Obj006	右	50,30
Obj007	右	50,30
Obj008	下	30,50
Obj009	右	50,30
Obj010	下	30,50

第14図 グラフィカル・オブジェクトのテーブル  
(ベース・テーブル)

オブジェクト・オペレーション・テーブル

オブジェクトID	オペレーション	属性ID
Obj001	マウス・クリック	Tr001
Obj002	マウス・クリック	Tr002
Obj003	マウス・クリック	Tr003
Obj004	マウス・クリック	Tr004
Obj005	マウス・クリック	Tr005

第15図 グラフィカル・オブジェクトと  
ユーザー・アクションの関係を表すテーブル  
(リレーションシップ・テーブル)

バキル定義テーブル

オブジェクトID	位置
Obj001	(30,300)
Obj002	(280,300)
Obj003	(280,150)
Obj004	(530,150)
Obj005	(530,50)
Obj006	(230,300)
Obj007	(480,300)
Obj008	(330,230)
Obj009	(480,150)
Obj010	(630,190)

第15図 画面を構成する  
グラフィカル・オブジェクトを記述するテーブル  
(リレーションシップ・テーブル)

手続キ起動テーブル

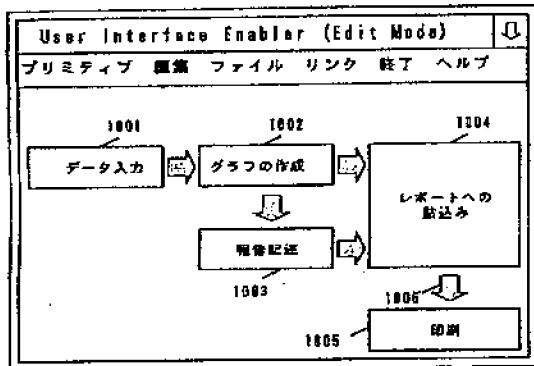
属性ID	起動手續ID
Tr001	Pro001
Tr002	Pro002
Tr003	Pro003
Tr004	Pro004
Tr005	Pro005

第17図 手続キを起動するためのテーブル

ウィンドウ・メッセージ・テーブル

ウィンドウ・ハンドル	メッセージID	パラメタ1	パラメタ2
W000	マウス・クリック	(300,350)	

第18図 ウィンドウ・メッセージのためのテーブル



第19図A 手続きパネル

印刷	
ファイル名	C : _
枚数	1 枚
開始ページ	0 ページ
終了ページ	0 ページ
<input type="button" value="実行"/> <input type="button" value="キャンセル"/>	

This is a configuration dialog for printing. It includes fields for file name (C : \_), number of copies (1), starting page (0), ending page (0), and two buttons at the bottom: '実行' (Execute) and 'キャンセル' (Cancel).

第19図B 印刷パネル

パネル・テーブル

パネルID	名前
Pan001	手書きパネル
Pan002	印刷パネル

手続き定義テーブル

手続ID	名前
Pro001	StarDate
Pro002	MakeGraph
Pro003	WriteRpt
Pro004	PasteRpt
Pro005	PrintRpt

第20図 パネル・テーブル、手続きモジュールのテーブル  
(ベース・テーブル)

パネル遷移テーブル

遷移ID	現在の画面	次の画面
Tr005	Pan001	Pan002

手続き遷移テーブル

遷移ID	起動手続き
Tr001	Pro001
Tr002	Pro002
Tr003	Pro003
Tr004	Pro004

第21図 パネルの遷移を指定するテーブル  
手続きを起動するためのテーブル